

Lesson 1 - Creating a Project

The goals for this lesson are:

- Create a project

A project is a collection entity for an HDL design under specification or test. Projects ease interaction with the tool and are useful for organizing files and simulation settings. At a minimum, projects have a work library and a session state that is stored in a .mpf file. A project may also consist of:

- HDL source files or references to source files
- other files such as READMEs or other project documentation
- local libraries
- references to global libraries

For more information about using project files, see the *ModelSim User's Manual*.

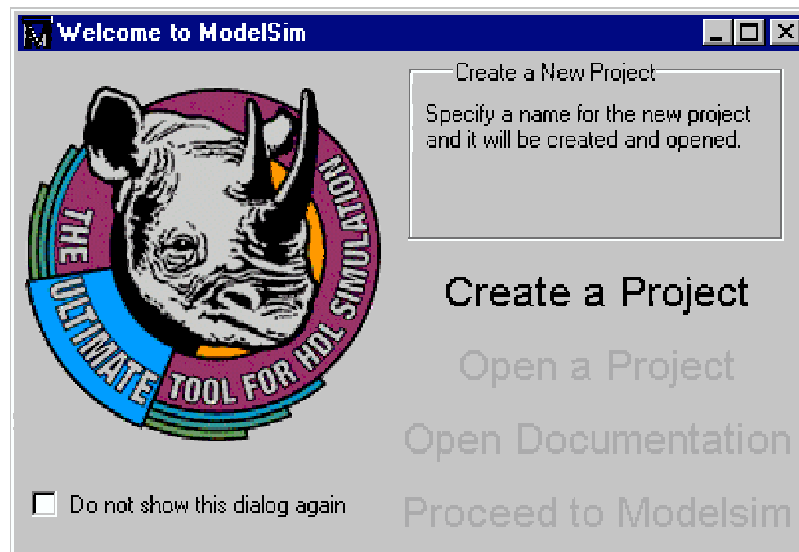
Creating a Project

- 1 Start ModelSim with one of the following:
for UNIX at the shell prompt:

```
vsim
```


for Windows - your option - from a Windows shortcut icon, from the Start menu, or from a DOS prompt:

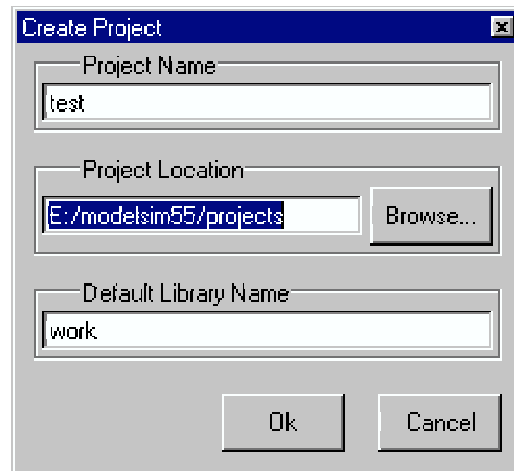
```
modelsim.exe
```
- 2 Upon opening ModelSim for the first time, you will see a Welcome to ModelSim dialog box. (If this screen is not available, you can enable it by selecting **Help > Enable Welcome** from the Main window. It will then display the next time you start ModelSim.)



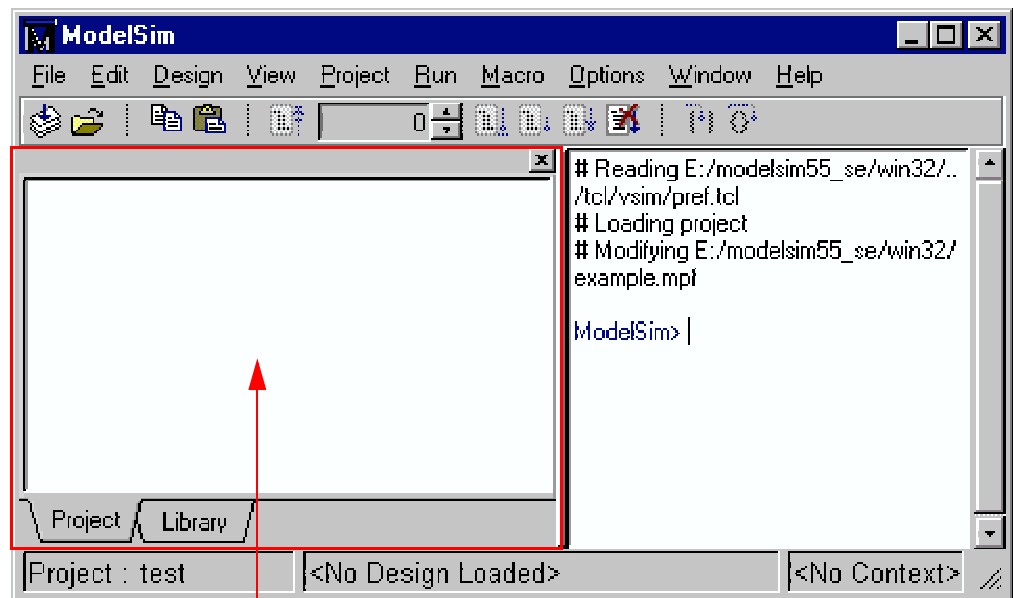
Select **Create a Project** from the Welcome to ModelSim dialog box. Or select **File > New > Project** from the ModelSim Main window.

Selecting **Create a Project** opens the Create Project dialog box.

- 3 In the Create Project dialog box, enter "test" as the Project Name and select a directory where the project file will be stored. Leave the Default Library Name set to "work."

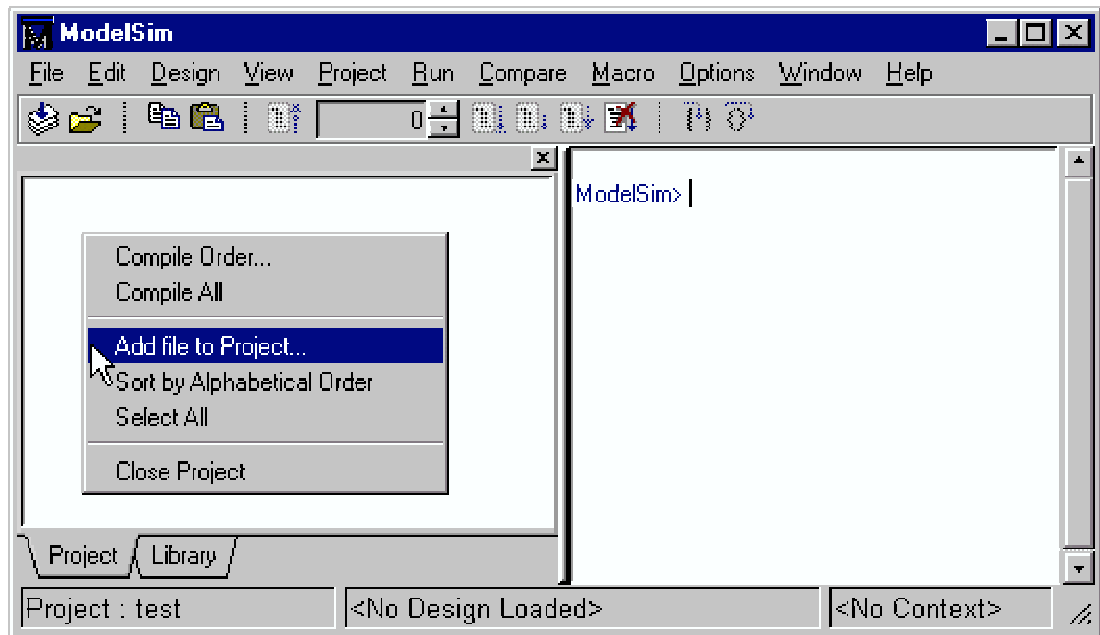


- 4 Upon selecting OK, you'll see the Main window with Project and Library tabs. Notice too that the Project name is shown in the status bar below the Workspace.

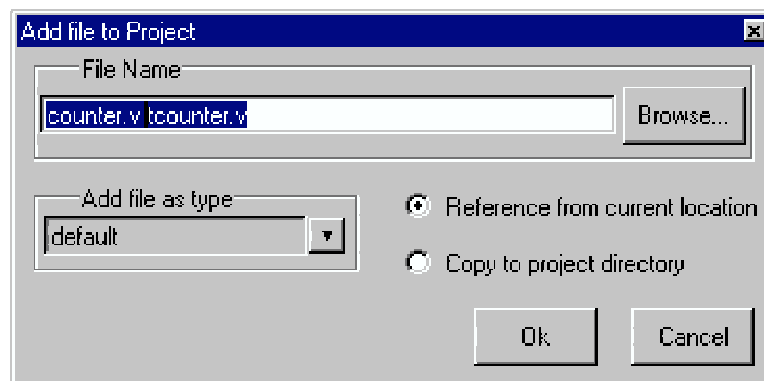


Workspace

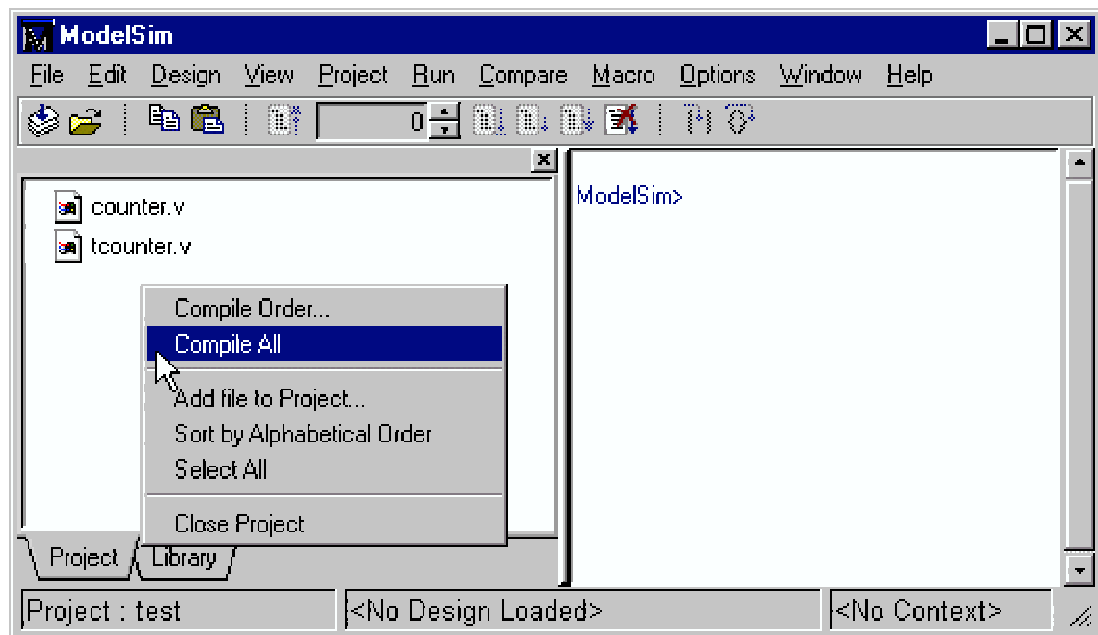
- 5 The next step is to add the files that contain your design units. Click your right mouse button (2nd button in Windows; 3rd button in UNIX) in the Project page of the Workspace, and select **Add File to Project**.



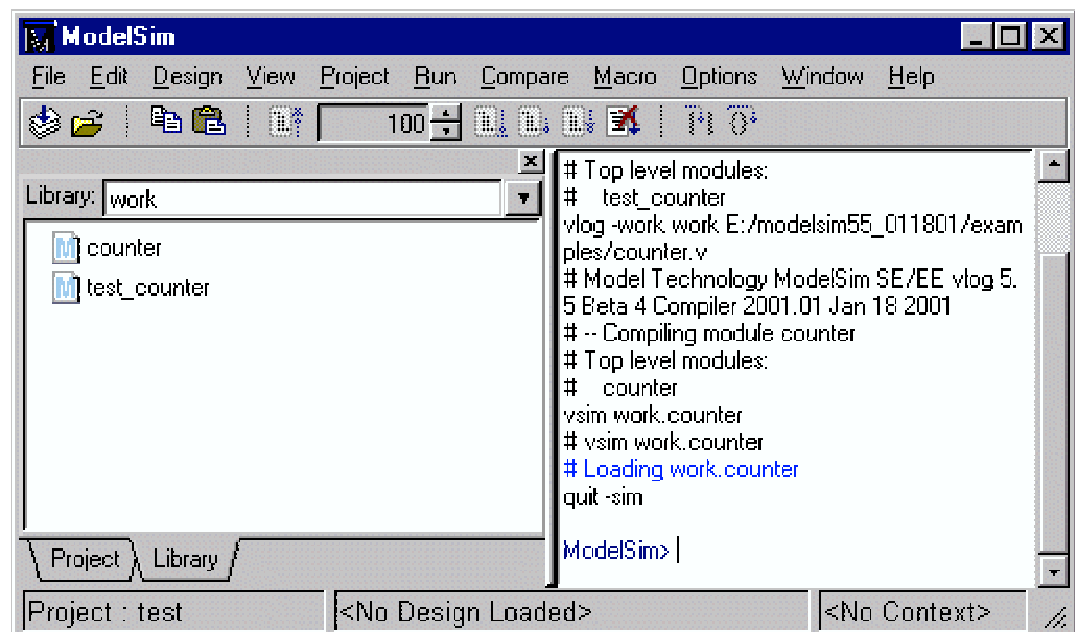
- 6 For this exercise, we'll add two Verilog files. Click the **Browse** button in the Add file to Project dialog box and open the examples directory in your ModelSim installation. Select *tcounter.v* and *counter.v*. Select **Reference from current location** and then click OK.



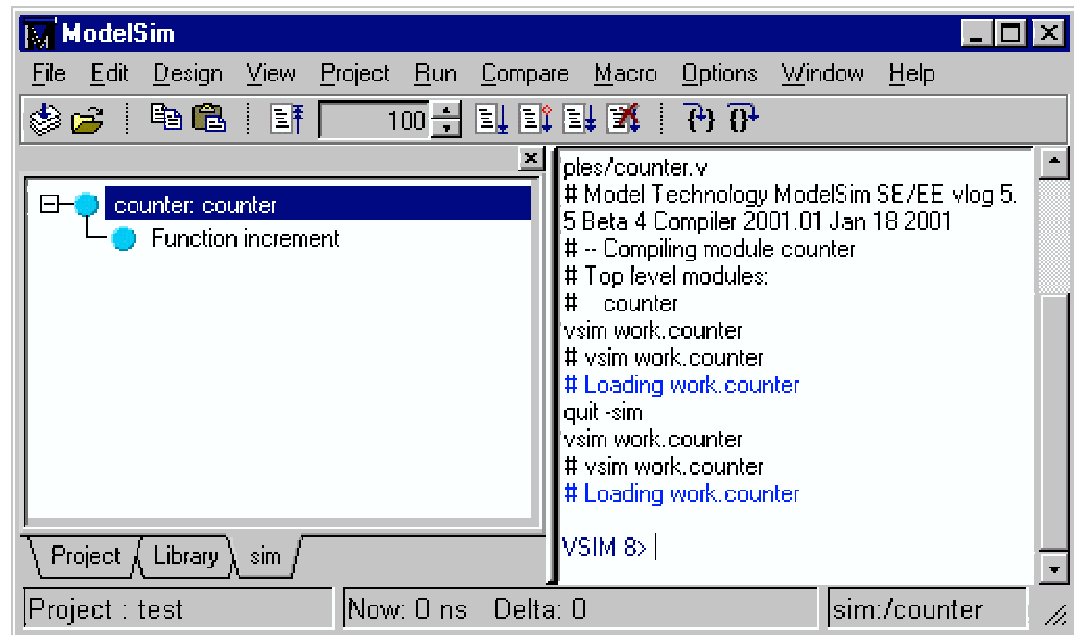
- 7 Right click in the Project page and select **Compile All**.



- 8 The two files are compiled. Click on the Library tab and you'll see the compiled design units listed. If you don't see the design units, make sure the Library: field shows "work."



- 9 The last step in this exercise is to load one of the design units. Double-click *counter* on the Designs page. You'll see a new page appear in the workspace that displays the structure of the *counter* design unit.



At this point, you would generally run the simulation and analyze or debug your design. We'll do just that in the upcoming lessons. For now, let's wrap up by ending the simulation and closing the project. Select **Design > End Simulation** and after confirming you want to quit simulating, select **File > Close > Project**.

Lesson 2 - Basic VHDL simulation

The goals for this lesson are:

- Create a library
- Compile a VHDL file
- Start the simulator
- Learn about the basic *ModelSim* windows, mouse, and menu conventions
- Run *ModelSim* using the **run** command
- List some signals
- Use the waveform display
- Force the value of a signal
- Single-step through a simulation run
- Set a breakpoint

The project feature covered in Lesson 1 executes several actions automatically such as creating and mapping work libraries. In this lesson we will go through the whole process so you get a feel for how *ModelSim* really works.

Preparing the simulation

- 1 Start by creating a new directory for this exercise (in case other users will be working with these lessons). Create the directory, then copy all of the VHDL (.vhd) files from `\<install_dir>\modeltech\examples` to the new directory.

Make sure the new directory is the current directory. Do this by invoking *ModelSim* from the new directory or by selecting the **File > Change Directory** command from the *ModelSim* Main window.

- 2 Start *ModelSim* with one of the following:

for UNIX at the shell prompt:

```
vsim
```

for Windows - your option - from a Windows shortcut icon, from the Start menu, or from a DOS prompt:

```
modelsim.exe
```

- ▶ **Note:** if you didn't add *ModelSim* to your search path during installation, you will have to include the full path when you type this command at a DOS prompt.

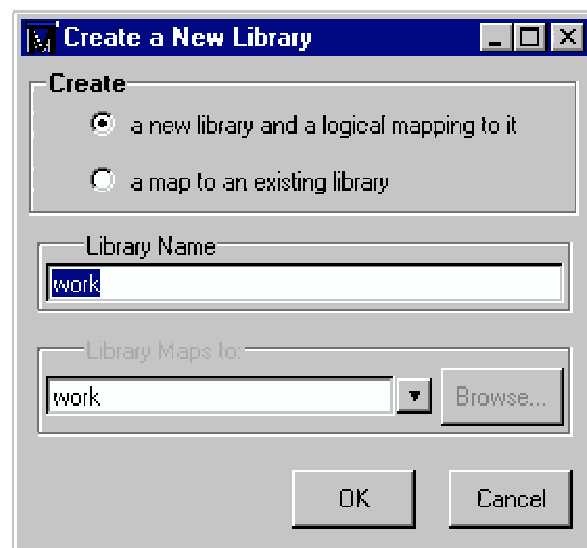
Select "Proceed to ModelSim" if the Welcome dialog appears.

- 3 Before you compile any HDL code, you'll need a design library to hold the compilation results. To create a new design library, make this menu selection in the Main window: **Design > Create a New Library**.

Make sure **Create: a new library and a logical mapping to it** is selected. Type "work" in the Library Name field and then select **OK**.

This creates a subdirectory named *work* - your design library - within the current directory. *ModelSim* saves a special file named *_info* in the subdirectory.

(PROMPT: vlib work
vmap work work)



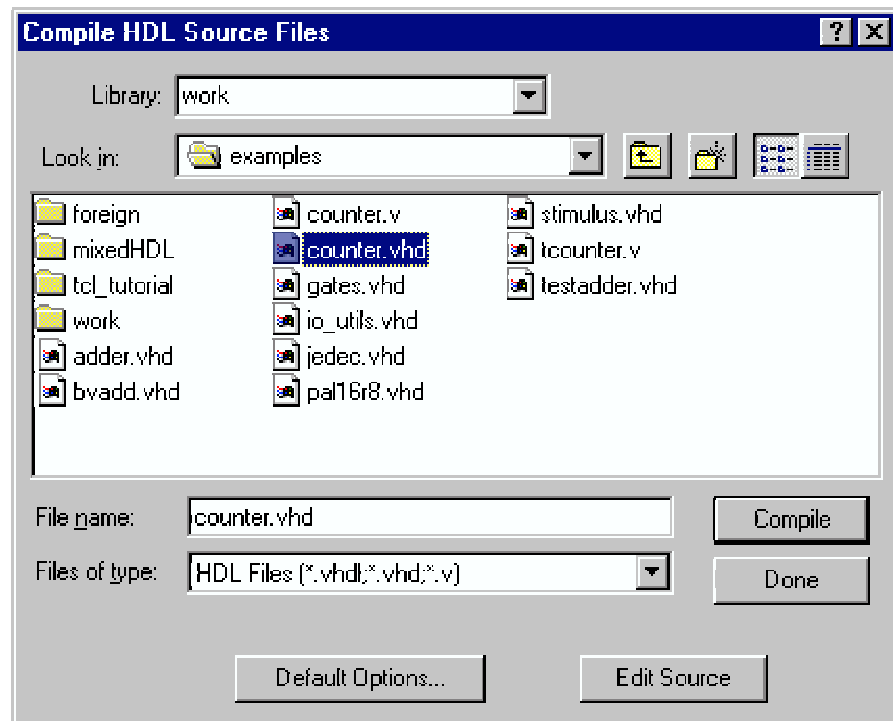
- ▶ **Note:** Do not create a Library directory using UNIX or Windows commands, because the *_info* file will not be created. Always use the Design menu or the **vlib** command from either the *ModelSim* or UNIX/DOS prompt.)

- 4 Compile the file *counter.vhd* into the new library by selecting the **Compile** button on the toolbar:



(PROMPT: vcom counter.vhd)

This opens the Compile HDL Source Files dialog box. (You won't see this dialog box if you invoke vcom from the command line.)



Complete the compilation by selecting *counter.vhd* from the file list and clicking **Compile**. Select **Done** when you are finished.

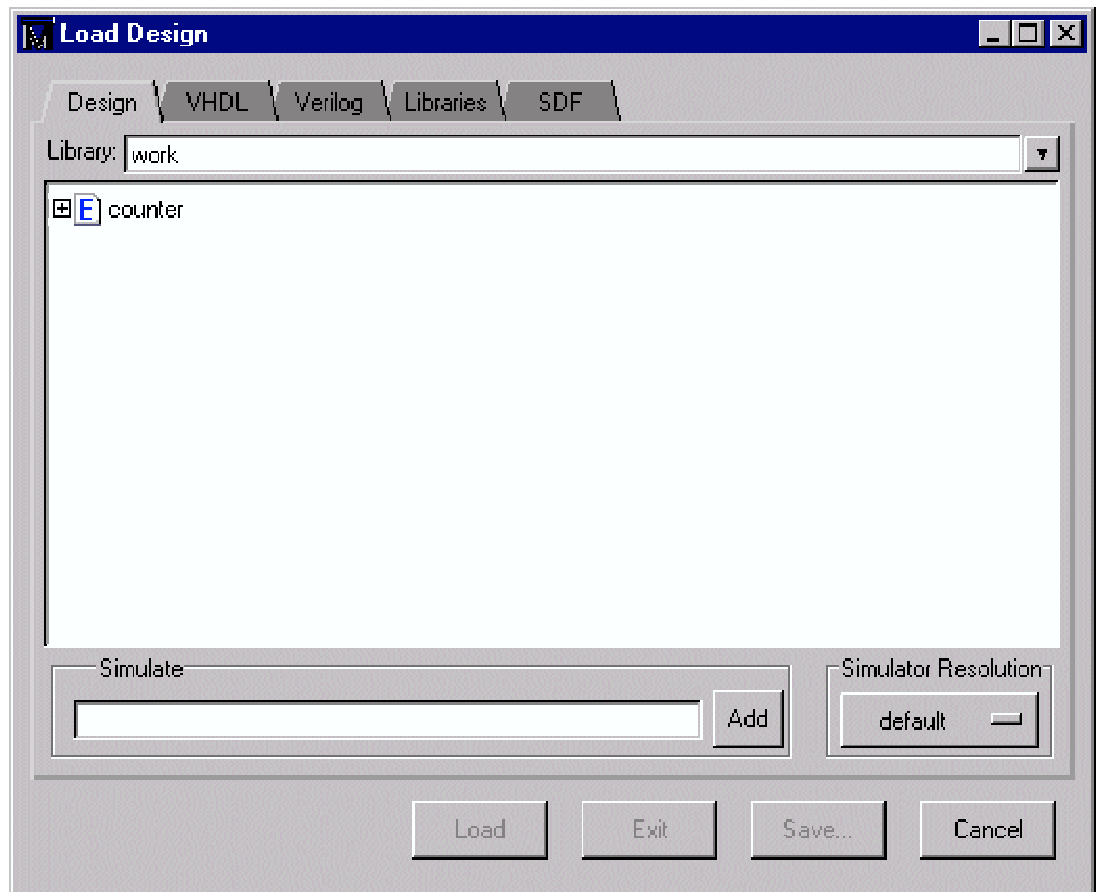
You can compile multiple files in one session from the file list. Individually select and Compile the files in the order required by your design.

- 5 Now let's load the design unit. Select the **Load Design** button from the toolbar:



(PROMPT: vsim counter)

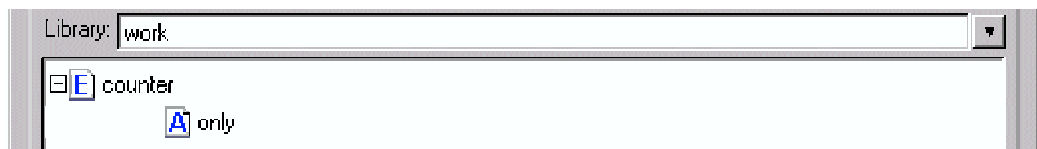
The Load Design dialog box comes up, as shown below (you won't see this dialog box if you invoke **vsim** with *counter* from the command line).



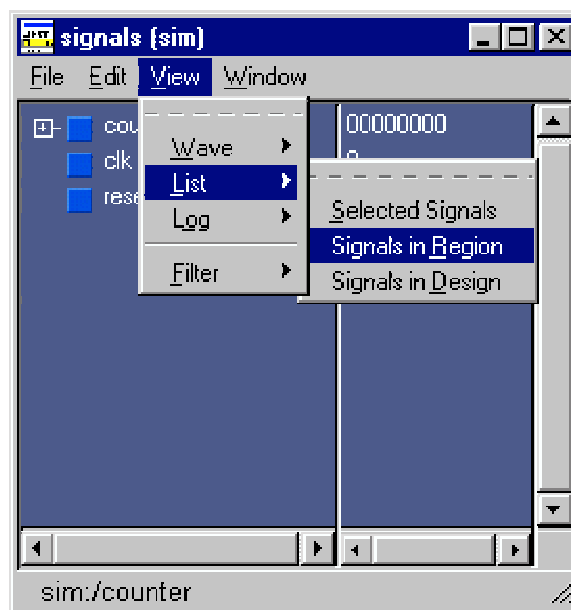
The Load Design dialog box lets you select the library and top-level design unit to simulate. You can also select the resolution limit for this simulation. By default, the following will appear for this simulation run:

- Simulator Resolution: default (the default is 1 ns)
- Library: work
- Design Unit: counter

If the Design Unit is an entity (like **counter** in this design), you can click on the plus-box prefix to view any associated architectures.



- 6 Select the entity **counter** and choose **Load** to accept these settings.
- 7 Next, select **View > All** from the Main window menu to open all ModelSim windows.
(PROMPT: view *)
For descriptions of the windows, consult the *ModelSim User's Manual*.
- 8 From the Signals window menu, select **View > List > Signals in Region**. This command displays the top-level signals in the List window.
(PROMPT: add list /counter/*)



- 9 Next add top-level signals to the Wave window by selecting **View > Wave > Signals in Region from the Signals** window menu.
(PROMPT: add wave /counter/*)

Running the simulation

We will start the simulation by applying stimulus to the clock input.

- 1 Click in the Main window and enter the following command at the VSIM prompt:

```
force clk 1 50, 0 100 -repeat 100
```

(MENU: Signals > Edit > Clock)

ModelSim interprets this **force** command as follows:

- force clk to the value 1 at 50 ns after the current time
- then to 0 at 100 ns after the current time
- repeat this cycle every 100 ns

- 2 Now you will exercise two different **Run** functions from the toolbar buttons on either the Main or Wave window. (The **Run** functions are identical in the Main and Wave windows.) Select the **Run** button first. When the run is complete, select **Run All**.



Run. This causes the simulation to run and then stop after 100 ns.

(PROMPT: run 100) (MENU: Run > Run 100ns)



Run -All. This causes the simulator to run forever. To stop the run, go on to the next step.

(PROMPT: run -all) (MENU: Run > Run -All)

- 3 Select the **Break** button on either the **Main** or **Wave** toolbar to interrupt the run. The simulator will stop running as soon as it gets to an acceptable stopping point.

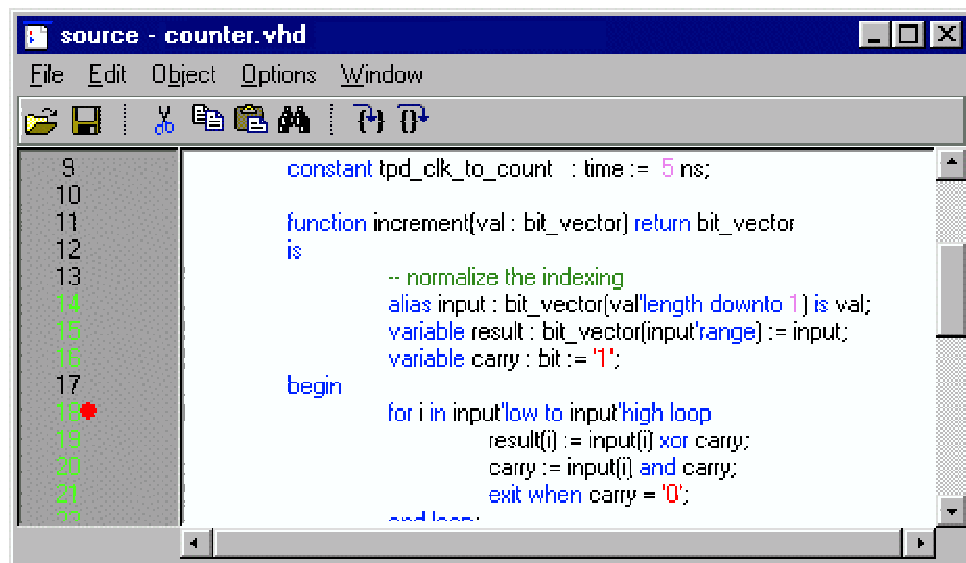


The arrow in the Source window points to the next HDL statement to be executed. (If the simulator is not evaluating a process at the time the Break occurs, no arrow will be displayed in the Source window.)

Next, you will set a breakpoint in the function on line 18.

- 4 Move the pointer to the Source window. Scroll the window vertically until line 18 is visible. Click on or near line number 18 to set the breakpoint. You should see a red dot next to the line number where the breakpoint is set. The breakpoint can be toggled between enabled and disabled by clicking it. When a breakpoint is disabled, the circle appears open. To delete the breakpoint, click the line number with your right mouse button and select Remove Breakpoint 18.

(PROMPT: bp counter.vhd 18)



- **Note:** Breakpoints can be set only on executable lines — denoted by green line numbers.

- 5 Select the **Continue Run** button to resume the run that you interrupted. ModelSim will hit the breakpoint, as shown by an arrow in the Source window and by a Break message in the Main window.



(PROMPT: run -continue) (MENU: Run > Continue)

- 6 Click the **Step** button to single-step through the simulation. Notice that the values change in the Variables window. You can keep clicking **Step** if you wish.



(PROMPT: run -step) (PROMPT: step)

- 7 When you're done, quit the simulator by entering the command:

```
quit -force
```

This command exits ModelSim without asking for confirmation.

